Jaeger

🕐 2 minute read ✔ page test

After completing this task, you understand how to have your application participate in tracing with Jaeger, regardless of the language, framework, or platform you use to build your application.

This task uses the Bookinfo sample as the example application.

To learn how Istio handles tracing, visit this task's overview.

Before you begin

1. Follow the `Jaeger installation` documentation to deploy Jaeger into your cluster.

2. When you enable tracing, you can set the sampling rate that Istio uses for tracing. Use the `meshConfig.defaultConfig.tracing.sampling` option during installation to `set the sampling rate`. The default sampling rate is 1%.

3. Deploy the `Bookinfo` sample application.

Accessing the dashboard

Remotely Accessing Telemetry Addons details how to configure access to the Istio addons through a gateway.

For testing (and temporary access), you may also use port-forwarding. Use the following, assuming you've deployed Jaeger to the `istio-system` namespace:

```
$ istioctl dashboard jaeger
```

Generating traces using the Bookinfo sample

1. When the Bookinfo application is up and running, access `http://$GATEWAY_URL/productpage` one or

more times to generate trace information.

To see trace data, you must send requests to your service. The number of requests depends on Istio's sampling rate. You set this rate when you install Istio. The default sampling rate is 1%. You need to send at least 100 requests before the first trace is visible. To send a 100 requests to the productpage service, use the following command:

```
$ for i in $(seq 1 100); do curl -s -o /dev/nul
l "http://$GATEWAY_URL/productpage"; done
```

2. From the left-hand pane of the dashboard, select productpage.default from the **Service** drop-down list and click **Find Traces**:
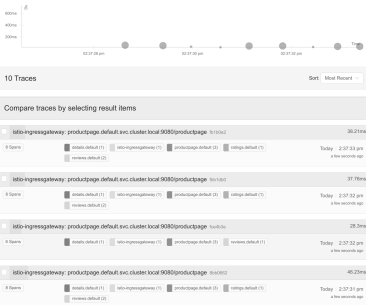
Tracing Dashboard

3. Click on the most recent trace at the top to see the details corresponding to the latest request to the /productpage:

4. The trace is comprised of a set of spans, where each span corresponds to a Bookinfo service, invoked during the execution of a `/productpage` request, or internal Istio component, for example: `istio-ingressgateway`.

Cleanup

1. Remove any `istioctl` processes that may still be running using control-C or:

   ```
   $ killall istioctl
   ```

2. If you are not planning to explore any follow-on tasks, refer to the Bookinfo

`cleanup` instructions to shutdown the application.